

# **Estudio Empírico de Operadores de Cruzamiento en un Algoritmo Genético Aplicado al Problema de Steiner Generalizado**

Martín Pedemonte  
CeCal, InCo  
Facultad de Ingeniería  
Universidad de la República  
Uruguay  
mpedemon@fing.edu.uy

Sergio Nesmachnow  
CeCal, InCo  
Facultad de Ingeniería  
Universidad de la República  
Uruguay  
sergion@fing.edu.uy

## **Resumen**

El Problema de Steiner Generalizado modela el diseño de redes de comunicaciones confiables en las cuales se exigen requisitos de conexión entre nodos distinguidos, que garanticen con alta probabilidad la comunicación entre sí. Es un problema NP difícil, para el cual pocos algoritmos han sido propuestos.

Los algoritmos evolutivos se han utilizado como metaheurísticas alternativas a los métodos exactos para resolver complejos problemas de optimización. Siguiendo este enfoque, hemos propuesto trabajar con algoritmos genéticos para la resolución del Problema de Steiner Generalizado.

La calidad de las soluciones obtenidas al utilizar un algoritmo genético depende de múltiples factores, entre los que se pueden destacar los operadores de recombinación y mutación utilizados. El objetivo de este trabajo es estudiar desde un punto de vista empírico la influencia de diferentes operadores de cruzamiento en un algoritmo genético para la resolución del Problema de Steiner Generalizado. Se presentan y comparan los resultados obtenidos utilizando cruzamientos de N puntos y variantes paramétricas del cruzamiento uniforme.

**Palabras clave:** Algoritmos genéticos, problema de Steiner generalizado, operadores de cruzamiento.

Destinado al Workshop de Agentes y Sistemas Inteligentes

## 1. Introducción

Uno de los principales problemas de construcción de redes de comunicaciones plantea el diseño de una topología de interconexión de sus nodos de modo que la red verifique ciertas características de confiabilidad. La confiabilidad de una red es una medida que evalúa la probabilidad de éxito en la comunicación entre pares de nodos.

El continuo crecimiento en el tamaño de los problemas de diseño de redes de comunicaciones ha conducido a proponer alternativas a los enfoques exactos tradicionales para resolver instancias complejas. En este contexto, varias heurísticas se han aplicado para obtener soluciones aproximadas de buena calidad en tiempos razonables. Entre ellas, los algoritmos genéticos (AG) se han manifestado como métodos flexibles y robustos para la solución de los complejos problemas de optimización subyacentes al diseño de redes de comunicaciones.

Dada una red de comunicaciones con nodos distinguidos denominados terminales, el Problema de Steiner Generalizado (GSP, por sus siglas en inglés) refiere al diseño de una subred de mínimo costo que verifique requerimientos prefijados de conexión entre pares de nodos distinguidos denominados terminales.

La minimización del costo total de una red de comunicaciones y la maximización de su confiabilidad son objetivos contrapuestos. Por ello, un modelo que minimice el costo total de la red satisfaciendo los requisitos de conexión sin agregar redundancia de caminos, constituye una solución muy sensible a fallas en los nodos o en los enlaces que afectarían la operatividad de la red.

El GSP incorpora requisitos adicionales de conectividad sobre pares de nodos terminales, aplicándose al diseño de redes de comunicaciones en las cuales la alta confiabilidad queda garantizada por la existencia de caminos alternativos entre nodos terminales. Esta redundancia otorga a la red un funcionamiento más robusto, siendo más resistente a fallas en sus componentes.

Tradicionalmente, los AG que utilizan codificación binaria simple han trabajado con los operadores de cruzamiento de uno y dos puntos, sin prestar especial atención a operadores que utilizan mayor número de puntos de corte. En nuestro entorno de trabajo hemos utilizado el cruzamiento de un punto en un AG para resolver el GSP [2], y en experimentos posteriores hemos utilizado cruzamiento de dos puntos y uniforme alcanzando mejoras en la calidad de los resultados obtenidos.

La propuesta de este trabajo consiste en estudiar desde un punto de vista empírico la influencia de diferentes operadores de cruzamiento en un AG para la resolución del GSP, de modo de determinar el operador más adecuado para el problema.

El resto del artículo se organiza del modo que se detalla a continuación. La sección 2 presenta el problema estudiado y su modelo matemático. La sección 3 describe el algoritmo genético utilizado y la codificación del problema. La sección 4 explica los operadores estudiados y su diseño. Los resultados experimentales sobre las instancias de prueba consideradas y las comparaciones entre variantes se presentan en la sección 5. Por último, la sección 6 ofrece las conclusiones y propuestas de trabajo futuro.

## 2. El Problema de Steiner Generalizado

Esta sección presenta la formulación y el modelo matemático del GSP, describe variantes del problema y concluye con un breve comentario sobre trabajos relacionados con la aplicación de técnicas evolutivas para la resolución del problema.

### 2.1 - Formulación y modelo matemático del GSP

La siguiente formulación del GSP se basa en el compendio de Kahn y Crescenzi [7]. Sean los siguientes elementos:

- Un grafo no dirigido  $G = (V, E)$ , siendo  $V$  el conjunto de nodos y  $E$  el conjunto de aristas, y con una matriz de costos  $C$  asociados a las aristas.
- Un conjunto distinguido de nodos  $T$  llamados *terminales*, de cardinalidad  $n_T = |T|$ , tal que  $2 \leq n_T \leq n_V$ , siendo  $n_V = |V|$  la cardinalidad del conjunto de vértices del grafo  $G$ .
- Una matriz  $R = \{r_{ij}\}$  con  $i, j \in T$ , de dimensión  $n_T \times n_T$ , cuyos elementos son enteros positivos que indican los requerimientos de conectividad –cantidad de caminos disjuntos– que se exigen entre todo par de nodos terminales.

El problema GSP propone encontrar un subgrafo  $G_T$  de  $G$ , de costo mínimo, tal que todo par de nodos  $i, j \in T$ ,  $i \neq j$  sean  $r_{ij}$  arista conexos en  $G_T$ , es decir que existan  $r_{ij}$  caminos disjuntos, que no comparten aristas, entre los nodos  $i$  y  $j$  en el grafo  $G_T$ .

Sobre los nodos no pertenecientes al conjunto de nodos terminales no se plantean requisitos de conectividad. Estos nodos, llamados *nodos de Steiner*, pueden o no formar parte de la solución óptima, de acuerdo a la conveniencia de utilizarlos.

Un modelo matemático del problema GSP arista conexo define para cada arista  $(i, j) \in E$  una variable binaria  $x_{ij}$  y una variable real  $y_{ij}^{kl}$  que representa la utilidad de la arista  $(i, j)$  en la dirección de  $i$  hacia  $j$  en un camino que une a los nodos terminales  $k$  y  $l$ .

El Cuadro 1 presenta el modelo matemático del GSP como problema de programación lineal entera, de acuerdo al trabajo de Robledo [9].

$$\begin{aligned}
 & \text{Min } \sum_{(i,j) \in E} C_{ij} \cdot x_{ij} \quad \text{sujeto a} \\
 & x_{ij} \geq y_{ij}^{kl} + y_{ji}^{kl} \quad \forall (i, j) \in E, \forall k, l \in T, k \neq l \\
 & \sum_{(k,l) \in E} y_{kj}^{kl} \geq r_{kl} \quad \forall k, l \in T, k \neq l \\
 & \sum_{(p,j) \in E} y_{pj}^{kl} - \sum_{(i,p) \in E} y_{ip}^{kl} \geq 0 \quad \forall k, l \in T, \forall p \in V \setminus \{k, l\} \\
 & x_{ij} \in \{0, 1\} \quad \forall (i, j) \in E \\
 & y_{ij}^{kl} \geq 0 \quad \forall i, j : (i, j) \in E, \forall k, l \in T, k \neq l
 \end{aligned}$$

*Cuadro 1: Modelo matemático del GSP como problema de programación lineal entera*

Llamando  $U = \{u_{ij}\}$  a una solución óptima del problema, el conjunto de aristas determinado por  $\{(i, j) \in E / u_{ij} = 1\}$  define un subgrafo solución del problema.

## 2.2 - Variantes y complejidad del problema

La complejidad del problema de Steiner está dada por la generalidad de su planteo, que exige requisitos variables de conexión entre pares de nodos terminales. Ciertas variantes del problema simplifican este requerimiento: los *problemas de  $k$  conexión* exigen una cantidad fija  $k$  de caminos disjuntos entre pares de nodos terminales, mientras que el *problema del árbol de Steiner* sólo exige la existencia de un camino entre pares de nodos terminales.

El GSP forma parte de la clase de problemas NP difíciles [7]. Karp probó en 1972 que el propio problema del árbol de Steiner, que plantea la restricción menos general de caminos, es NP completo [8]. La complejidad del problema de Steiner lo hace cada vez menos tratable al aumentar el tamaño del grafo involucrado. El uso de algoritmos exactos queda limitado a pequeñas instancias de complejidad reducida, como consecuencia de los elevados tiempos de ejecución necesarios para su resolución. Por tal motivo se han propuesto soluciones alternativas utilizando heurísticas que permitan encontrar soluciones de buena calidad en tiempos razonables.

### 2.3 - Técnicas evolutivas aplicadas a la resolución del problema de Steiner

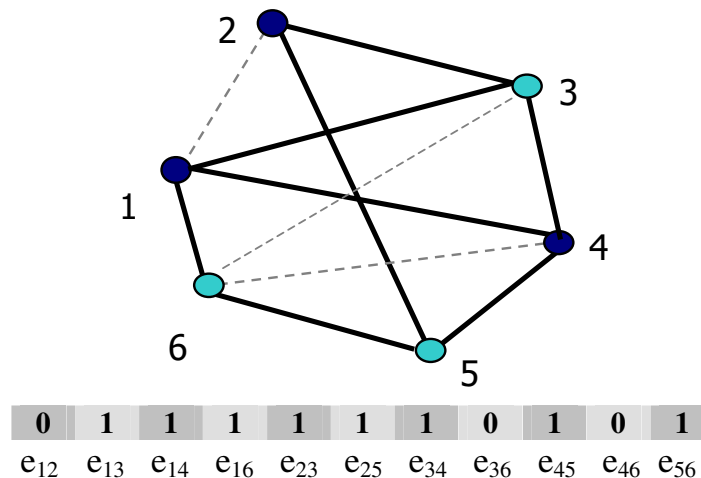
No existen antecedentes sobre la resolución del GSP mediante técnicas evolutivas. Existen trabajos sobre AG aplicados al problema del árbol de Steiner, pero fuera de nuestro entorno de trabajo no hay referencias de aplicación de técnicas evolutivas al problema generalizado. El enfoque de este trabajo corresponde al presentado en un artículo precedente [2] donde se ha seguido una propuesta de Esbensen [4] para el problema del árbol de Steiner, imponiendo el trabajo con soluciones factibles.

## **3. Implementación mediante un Algoritmo Genético**

Esta sección presenta los detalles de codificación e implementación del AG utilizado para la resolución del GSP. Los operadores de cruzamiento constituyen la parte fundamental de nuestro estudio y son presentados en detalle en la sección siguiente.

### 3.1 - Codificación y operadores

El algoritmo genético implementado se basa en una codificación binaria sencilla. Cada cromosoma consta de un arreglo de bits que representan a las aristas del grafo original, numeradas de 0 a número de aristas - 1. La presencia o ausencia de una arista en un grafo solución queda determinada por el valor 1 o 0 respectivamente, en la posición correspondiente en el cromosoma. La Figura 1 presenta un ejemplo de la codificación binaria utilizada para un grafo pequeño, donde los nodos terminales se colorean en oscuro y los nodos de Steiner en claro. Las aristas marcadas en negro son aquellas presentes en el grafo representado y se notan con el valor 1 en el cromosoma de bits. Las aristas marcadas con la línea punteada gris están presentes en el grafo original pero no en el grafo solución representado, por lo cual se codifican con el valor 0 en el cromosoma. Las etiquetas de las posiciones del cromosoma referencian a los nodos origen y destino de cada arista del grafo original ( $e_{25}$  = arista entre nodo 2 y nodo 5).



*Figura 1 – Ejemplo de la codificación binaria utilizada*

Al utilizar la representación binaria los operadores de cruzamiento o mutación resultan sencillos, pero tienen el inconveniente de que los cromosomas resultantes pueden no representar a una solución factible del GSP. Por tal motivo, es necesario verificar la factibilidad de los nuevos cromosomas creados al aplicar un operador evolutivo.

En nuestra implementación hemos adoptado el criterio de descartar individuos no factibles. Esta decisión evita la tarea de cuantificar cuan lejos de una solución factible se encuentra un individuo no factible y determinar un valor adecuado de penalización para su fitness. Esta idea sigue la propuesta de Esbensen en su trabajo sobre el problema del árbol de Steiner [4] y ya ha sido utilizada en el trabajo previo [2].

Los operadores utilizados por el AG propuesto corresponden a:

- Selección por torneo entre tres individuos.
- Mutación de inversión del valor de un alelo, seleccionado probabilísticamente.
- Inicialización aleatoria de la población, mediante un procedimiento de eliminación de aristas a partir del grafo original.

Hemos trabajado con un criterio de parada de esfuerzo prefijado en 2000 generaciones, para realizar una comparación justa entre los resultados de los diferentes operadores de cruzamiento. Usar otros criterios, por ejemplo que analicen la evolución de fitness promedio, podría conducir a comparaciones injustas para operadores que introduzcan menor variabilidad entre generaciones pero evolucionen favorablemente a largo plazo.

### 3.2 - Función de fitness

La función de fitness utilizada toma en cuenta el costo del grafo representado por un cromosoma. Sustrayendo el valor de costo de un individuo del valor del costo del grafo original es posible mapear el problema de minimización del costo del grafo en el problema de maximización de la función presentada en la Ecuación 1.

$$f = C_{ORIG} - \sum_{i=0}^{|E|-1} [EDGE(i) * C(i)]$$

*Ecuación 1: Función de fitness para el problema GSP.*

En la Ecuación 1,  $|E|$  denota la cardinalidad del conjunto de aristas del grafo original, correspondiente al largo de la codificación utilizada.  $C_{ORIG}$  representa el costo del grafo original, donde todas las aristas se consideran presentes. La función  $C: [0, |E|-1] \rightarrow R$  devuelve el costo de una arista y la función  $EDGE: [0, |E|-1] \rightarrow [0, 1]$  retorna el valor binario correspondiente a la arista que ocupa la posición  $i$ -ésima en el cromosoma.

### 3.3 - Cálculo de factibilidad

La decisión de trabajar con individuos factibles simplifica el diseño de los operadores evolutivos, pero implica utilizar un procedimiento adicional para determinar la factibilidad de individuos luego de aplicar un operador. El cálculo de factibilidad verifica que el grafo representado cumpla las restricciones del GSP.

El algoritmo para cálculo de factibilidad consta de dos etapas:

- Una sencilla heurística se utiliza para detectar soluciones no factibles, verificando que los grados de los nodos terminales sean mayores o iguales al número máximo de caminos impuestos como restricción para cada nodo. La complejidad de esta heurística es de orden cuadrático en la cardinalidad del conjunto de nodos terminales y evita el cálculo de caminos entre pares de nodos terminales en todos los casos.
- Que un grafo pase el chequeo heurístico de grados de nodos terminales no implica que sea solución factible del GSP. En tal caso deben hallarse los caminos entre cada par de nodos terminales. Hemos utilizado una variante del algoritmo de Ford–Fulkerson [5] para cálculo de flujo máximo, que permite hallar caminos entre pares de nodos terminales asignando a un nodo el rol de fuente y al otro el rol de pozo. El algoritmo tradicional de Ford–Fulkerson trabaja sobre grafos dirigidos, por lo cual en el caso del grafo del GSP cada arista del grafo se considera como un par de aristas de sentidos opuestos. Asumiendo capacidad unitaria para las aristas, el flujo máximo entre fuente y pozo coincide con el número máximo de caminos disjuntos entre ellos. Si éste es menor que el valor del requerimiento correspondiente, el grafo es no factible. La variante de Ford–Fulkerson descrita utiliza una función de capacidad de flujo unitaria para cada arista, y tiene orden  $O(|E|.r_{MAX}.n_T^2)$ , siendo  $|E|$  la cardinalidad del conjunto de aristas del grafo original,  $r_{MAX}$  el valor máximo del conjunto de requerimientos, y  $n_T$  la cardinalidad del conjunto de nodos terminales.

### 3.4 - Implementación y plataforma de ejecución

El algoritmo genético fue implementado sobre MALLBA [1], una biblioteca para optimización que implementa una gama de algoritmos evolutivos puros e híbridos en sus versiones secuenciales y distribuidas. MALLBA está implementada mediante clases C++ que abstraen a las entidades participantes en la resolución de problemas. Existen *Clases Provistas* que implementan los esqueletos de los métodos de resolución de manera independiente del problema específico a resolver y *Clases Requeridas* que contienen las entidades dependientes del problema y deben ser instanciadas por el usuario. La propia biblioteca resuelve la interacción entre el método de resolución y el problema instanciado en las clases provistas por el usuario.

La biblioteca MALLBA fue desarrollada por las universidades de Málaga, La Laguna y Barcelona y está disponible en <http://neo.lcc.uma.es/mallba/easy-mallba>. En este trabajo hemos extendido las clases provistas de MALLBA implementando los operadores de cruzamiento que se presentan en la siguiente sección.

La ejecución de las pruebas se realizó sobre un cluster de equipos Intel Pentium 4 a 2.4 GHz, cada uno con 512 Mb RAM, con el sistema operativo SuSE Linux 8.0.

## **4. Operadores de Cruzamiento Utilizados**

Esta sección presenta la justificación del enfoque empírico del trabajo y las características de los operadores de cruzamiento considerados en el estudio.

### 4.1 - Motivación del estudio empírico

En el primer trabajo donde planteamos la resolución del GSP con un AG en versiones serial y paralela [2] no se realizó un análisis de operadores de cruzamiento, limitándose a trabajar con el cruzamiento de un punto. En experimentos posteriores utilizamos el cruzamiento de dos puntos y el uniforme, obteniendo mejoras en los resultados que nos condujeron a formalizar el estudio que se presenta en este artículo.

La complejidad del GSP y los aspectos intrínsecos de la codificación utilizada hacen muy complejo el análisis teórico de la influencia de los operadores de cruzamiento en la calidad de los resultados del AG. Las características de los esquemas que conviene privilegiar dependen de las particularidades del grafo que representa la red para la que se plantea el GSP. Tomando en cuenta la complejidad de diseño de redes de comunicaciones de mediano tamaño, no es sencillo explotar particularidades del problema que permitan determinar a priori un operador de cruzamiento adecuado.

Considerando la dificultad del análisis desde el punto de vista teórico, hemos utilizado un enfoque experimental para abordar el estudio de la influencia de los operadores de cruzamiento sobre la calidad de resultados de un AG para resolver el GSP.

### 4.2 - Operadores considerados en el estudio

No hay evidencia teórica o experimental que indique la existencia de un operador de cruzamiento universal que obtenga los mejores resultados para todo problema, siendo necesario determinar qué operador se ajusta a las características particulares de cada problema y de cada codificación para obtener buenos resultados.

Si bien existe una amplia gama de operadores de cruzamiento para AG, generalmente se han utilizado tres operadores de implementación sencilla, que trabajan sobre las codificaciones y no utilizan información del problema a resolver: el cruzamiento de un punto (1PX), el cruzamiento de dos puntos (2PX) y el cruzamiento uniforme (UPX). Este artículo presenta el análisis de estos operadores tradicionales y algunas variantes sencillas. Operadores más sofisticados, que utilizan estrategias adaptativas y adquieren cierto conocimiento del problema para mejorar su desempeño [10], no han sido considerados en este trabajo, proponiéndose su estudio para una etapa futura.

### 4.3 - Familia de operadores de cruzamiento de $n$ puntos

Esta familia de operadores propone definir puntos de corte e intercambiar segmentos entre los individuos padre. Su propuesta se remonta a los trabajos iniciales sobre AG donde se probaron resultados teóricos y empíricos que popularizaron su uso con valores bajos de  $n$ , generalmente 1 ó 2 [3,6]. Trabajos posteriores profundizaron el análisis teórico de esta familia de operadores, determinando cotas para la probabilidad de supervivencia para esquemas de cualquier orden. Al analizar las curvas de supervivencia en función del largo de definición para esquemas de un orden dado, Spears y De Jong [11] observaron que cuando  $n$  es par la disrupción crece al aumentar el número de puntos de corte. Cuando  $n$  es impar, para todo par de curvas de supervivencia existe un único punto de intersección, antes del cual, la disrupción crece al aumentar el número de puntos de corte. Luego de ese punto, las curvas modifican sus valores, transformándose en más disruptiva la estrategia que menos lo era y viceversa. Al considerar esquemas de mayor orden, los puntos de intersección de las curvas se desplazan hacia valores mayores, minimizando la zona en la cual se invierte el comportamiento de disrupción. Para esta familia de operadores la probabilidad de supervivencia depende fuertemente del largo de definición del esquema considerado. En este trabajo limitaremos el estudio de esta familia a los operadores de cruzamiento de un punto (1PX), dos puntos (2PX), tres puntos (3PX) y cuatro puntos (4PX). Para esquemas de bajo largo de definición los operadores pueden ordenarse trivialmente de acuerdo a su nivel de disrupción de menor a mayor: 1PX, 2PX, 3PX y 4PX.

### 4.4 - Familia de operadores de cruzamiento uniforme parametrizados

Esta familia de operadores propone el intercambio aleatorio de bits entre los individuos padre dependiendo de una probabilidad fija  $p$ . Las variantes paramétricas del operador se obtienen considerando  $p$  entre 0 y 0.5, de acuerdo a la simetría del problema. El operador UPX tradicional corresponde a utilizar un valor de  $p = 0.5$ .

El UPX fue propuesto por Syswerda [13], quien probó resultados teóricos acotando la probabilidad de supervivencia de esquemas y realizó estudios empíricos sobre la calidad de resultados al utilizar el operador para varios problemas de optimización. La probabilidad de supervivencia de esquemas bajo el operador UPX no depende del largo de definición, sino que es constante para todos los largos posibles. Posteriormente, Spears y DeJong plantearon una variante a la mecánica básica del UPX, estudiando el efecto de la parametrización de la probabilidad de intercambio entre los padres [12]. Esta variante mantiene las características del operador original, con probabilidad de supervivencia constante para los esquemas, pero al considerar parametrizaciones con valores menores de  $p$ , el cruzamiento resulta menos disruptivo que el UPX tradicional. En este trabajo limitaremos el estudio a los operadores de cruzamiento uniforme parametrizados con 0.1 (UP1X), 0.2 (UP2X), 0.3 (UP3X), 0.4 (UP4X), y 0.5 (usaremos el nombre UP5X en lugar de UPX, para homogeneizar la nomenclatura).

## **5. Resultados obtenidos**

Esta sección presenta las instancias del GSP sobre las que se evaluaron los operadores de cruzamiento, la parametrización del AG, los resultados obtenidos y su análisis.

### 5.1 - Conjunto de problemas de prueba

El GSP ha sido poco abordado por la comunidad científica, y como consecuencia no existen conjuntos de problema de prueba estandarizados para comparar la calidad de resultados de los AG. En nuestro entorno de trabajo hemos obtenido resultados sobre un conjunto de tres problemas cuyas topologías, número de nodos terminales y requerimientos consideramos representativos para redes de mediano tamaño.

Los grafos de prueba se generaron seleccionando aleatoriamente puntos en el plano euclídeo para representar los vértices e incluyendo aristas con probabilidad inversamente proporcional a la distancia euclidean entre los nodos que conectan. Las características de los tres grafos utilizados se resumen en la Tabla 1, indicando número total de vértices, número de terminales y número de aristas. Los grafos son designados por un nombre que denota la cantidad total de nodos y la cantidad de nodos terminales. El conjunto de grafos de prueba, y una descripción del formato utilizado se encuentran disponibles públicamente en <http://www.fing.edu.uy/cecal/hpc/gsp/TestSuite>. El generador aleatorio utilizado para crear los grafos también se halla disponible públicamente en <http://www.fing.edu.uy/cecal/hpc/gsp/Generator>.

	grafo 100-10	grafo 75-25	grafo 50-15
<b>Vértices</b>	100	75	50
<b>Terminales</b>	10	25	15
<b>Aristas</b>	500	360	249

Tabla 1: Características del conjunto de grafos de prueba.

### 5.2 - Parametrización del AG

Los parámetros del AG fueron fijados para estudiar en forma aislada el efecto de los operadores de cruzamiento sobre la calidad de los resultados. Se utilizaron poblaciones de 120 individuos, que evolucionaron durante 2000 generaciones. Las probabilidades de cruzamiento y de mutación fueron fijadas en 0.95 y 0.01 respectivamente.

### 5.3 - Resultados

Las tablas 2, 3 y 4 presentan los resultados promedio de 30 ejecuciones realizadas para cada grafo y para cada operador. Las tablas se dividen en tres secciones:

- *Mejores resultados obtenidos* detalla los valores para los mejores individuos alcanzados en las ejecuciones.
- *Características de la población final* detalla los resultados para el individuo promedio de la última generación.
- *Esfuerzo requerido para alcanzar el mejor valor obtenido* ofrece el número de generación y el tiempo necesario para obtener el mejor resultado e incluye como referencia el tiempo total de ejecución del algoritmo.

El estudio de eficiencia no es un objetivo de este trabajo, pero hemos incluido los tiempos de ejecución para detectar si eventuales mejoras en la calidad de resultados se logran a costa de un mayor tiempo de cómputo al utilizar una familia de operadores.

	<i>UPIX</i>	<i>UP2X</i>	<i>UP3X</i>	<i>UP4X</i>	<i>UP5X</i>	<i>IPX</i>	<i>2PX</i>	<i>3PX</i>	<i>4PX</i>
<i>Mejores resultados obtenidos</i>									
<i>Promedio</i>	4441	4474	4485	4497.5	4500	4457	4470	4477.5	4477
<i>Mejor</i>	4491	4569	4540	4548	4573	4502	4532	4521	4520
<i>Desv. Std.</i>	33.71	34.96	32.36	28.77	27.68	40.89	32.27	40.27	30.70
<i>Características de la población final</i>									
<i>Promedio</i>	4415.80	4446.22	4457.67	4470.51	4469.28	4428.22	4441.63	4444.75	4450.15
<i>Mejor</i>	4465.39	4540.07	4511.64	4523.31	4532.43	4477.64	4497.97	4478.97	4491.03
<i>Desv. Std.</i>	34.91	34.98	33.92	29.20	26.57	41.16	30.90	39.00	31.42
<i>Esfuerzo requerido para alcanzar el mejor valor obtenido</i>									
<i>Generación</i>	1756	1484.5	1456	1456	1593.5	1861	1800	1817	1725
<i>Tiempo (min.)</i>	112.93	95.58	93.48	94.95	98.20	114.93	108.68	109.47	106.16
<i>Tiempo total (min.)</i>	123.72	120.14	121.07	117.30	117.08	122.90	121.63	120.89	120.65

Tabla 2: Resultados para el grafo de 100 nodos y 10 terminales.



Las figuras 2 y 3 presentan gráficos con los resultados comparativos para cada familia de operadores para el grafo de 100 nodos y 10 terminales.

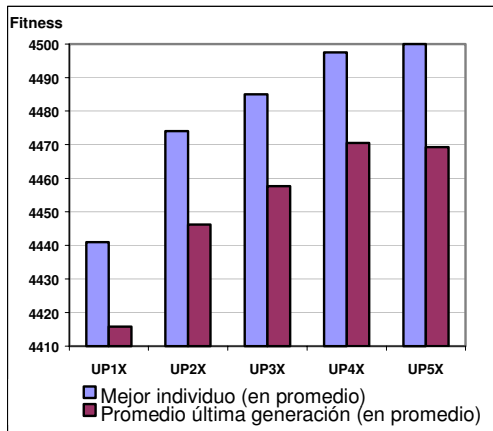


Figura 2: Resultados de operadores uniforme parametrizados, grafo 100-10.

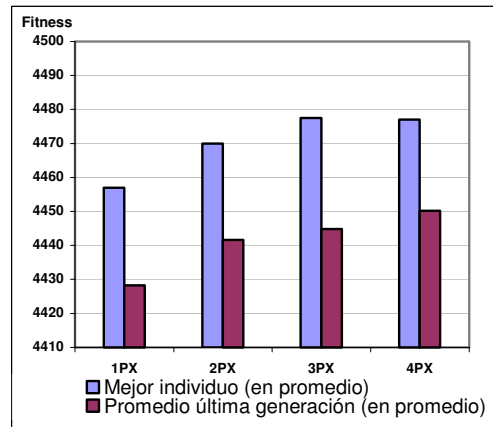


Figura 3: Resultados de operadores de  $n$  puntos, grafo 100-10.

En ambas familias de operadores las estrategias más disruptivas obtienen los mejores resultados, existiendo una concordancia entre el orden de operadores al considerar el mejor individuo y el promedio de la última generación. El desempeño de la familia de operadores uniformes parametrizados es notoriamente superior en calidad de los resultados obtenidos a los operadores de  $n$  puntos. Los mejores resultados para el grafo de 100 nodos y 10 terminales son alcanzados por los operadores UP5X y UP4X.

	UP1X	UP2X	UP3X	UP4X	UP5X	1PX	2PX	3PX	4PX
<i>Mejores resultados obtenidos</i>									
<i>Promedio</i>	5316.56	5341.36	5327.44	5350.67	5364.58	5290.98	5321.27	5321.5	5334.54
<i>Mejor</i>	5418.02	5415.41	5433.23	5465.85	5393.96	5374.11	5404.72	5388.13	5411.72
<i>Desv. Std.</i>	43.01	39.19	40.29	48.36	27.27	47.66	49.58	34.00	48.93
<i>Características de la población final</i>									
<i>Promedio</i>	5288.1	5304.58	5294.9	5320.78	5331.34	5254.16	5290.9	5286.75	5303.43
<i>Mejor</i>	5378.06	5382.83	5399.43	5441.19	5363.7	5350.71	5376.18	5368.08	5371.98
<i>Desv. Std.</i>	43.90	37.42	39.49	52.43	30.47	50.09	51.00	35.58	48.26
<i>Esfuerzo requerido para alcanzar el mejor valor obtenido</i>									
<i>Generación</i>	1795.5	1770.5	1404.5	1673	1543	1769.5	1792.5	1544	1736.5
<i>Tiempo (min.)</i>	288.07	279.17	228.93	262.96	253.56	279.57	271.27	253.30	273.10
<i>TiempoTotal (min.)</i>	313.12	306.12	308.42	309.42	316.43	314.93	303.00	312.80	313.30

Tabla 3: Resultados para el grafo de 75 nodos y 25 terminales.

Las figuras 4 y 5 presentan gráficos con los resultados comparativos para cada familia de operadores para el grafo de 75 nodos y 25 terminales.

Para el grafo de 75 nodos y 25 terminales, se reitera la superioridad en la calidad de los resultados de la familia de operadores uniformes parametrizados sobre la de  $n$  puntos. La relación directa entre la disruptción y la calidad de resultados al utilizar operadores de cruzamiento uniforme es violada por el buen desempeño del operador UP2X. En la familia de operadores de  $n$  puntos, el 2PX muestra buenos resultados, de calidad similar a los de 3PX. El operador 4PX obtiene los mejores resultados respecto al resto, pero las diferencias no son significativas, estando por debajo de la desviación estándar de las ejecuciones realizadas. Se mantiene la concordancia entre el orden de operadores al considerar el mejor individuo y el promedio de la última generación.

Nuevamente, los operadores UP5X y UP4X obtienen los mejores resultados.

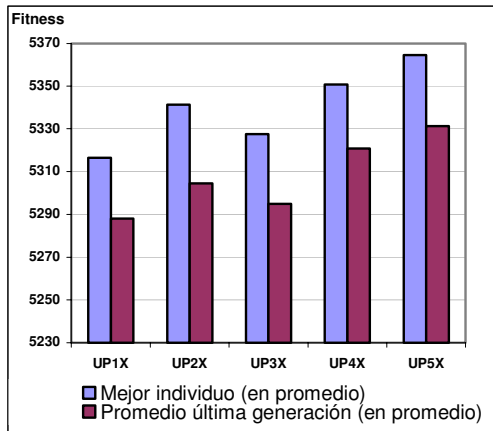


Figura 4: Resultados de operadores uniforme parametrizados, grafo 75-25.

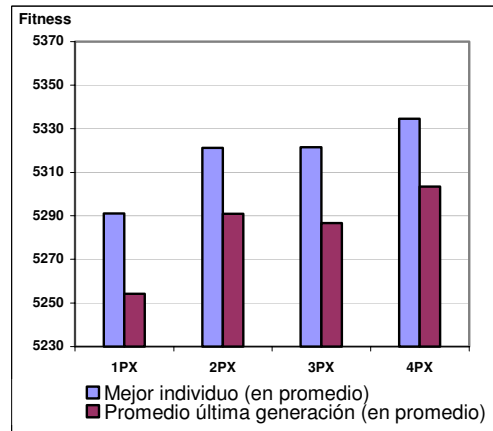


Figura 5: Resultados de operadores de n puntos, grafo 75-25.

	UPIX	UP2X	UP3X	UP4X	UP5X	IPX	2PX	3PX	4PX
<i>Mejores resultados obtenidos</i>									
<i>Promedio</i>	9301.63	9306.43	9333.71	9313.69	9347.36	9278.61	9283.62	9316.53	9344.58
<i>Mejor</i>	9494.02	9462.86	9454.77	9484.10	9564.75	9435.80	9464.56	9493.84	9520.01
<i>Desv. Std.</i>	97.95	79.66	95.22	66.13	81.96	72.22	77.46	89.49	79.19
<i>Características de la población final</i>									
<i>Promedio</i>	9235.17	9236.39	9262.97	9243.34	9267.19	9182.69	9206.88	9229.37	9249.08
<i>Mejor</i>	9413.57	9343.15	9400.79	9395.70	9466.82	9362.55	9404.81	9422.81	9437.19
<i>Desv. Std.</i>	101.40	75.76	95.02	69.78	82.14	73.62	83.08	90.47	82.06
<i>Esfuerzo requerido para alcanzar el mejor valor obtenido</i>									
<i>Generación</i>	1629.50	1673.50	1785.50	1421.00	1611.50	1570.50	1594.50	1633.00	1652.50
<i>Tiempo (min.)</i>	101.58	106.21	109.59	86.51	101.03	100.91	99.83	101.27	103.19
<i>TiempoTotal (min.)</i>	121.57	121.87	123.21	121.73	120.82	124.29	122.32	123.32	121.80

Tabla 4: Resultados para el grafo de 50 nodos y 15 terminales.

Las figuras 6 y 7 presentan gráficos con los resultados comparativos para cada familia de operadores para el grafo de 50 nodos y 15 terminales.

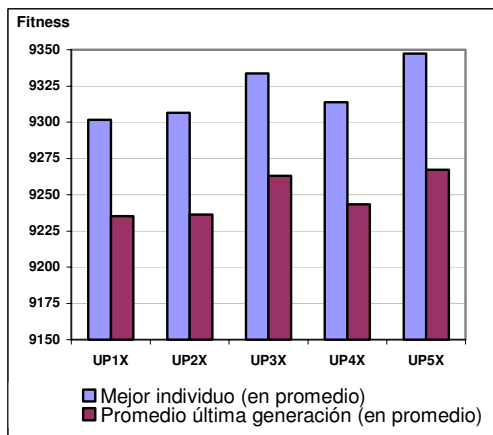


Figura 6: Resultados de operadores uniforme parametrizados, grafo 50-15.

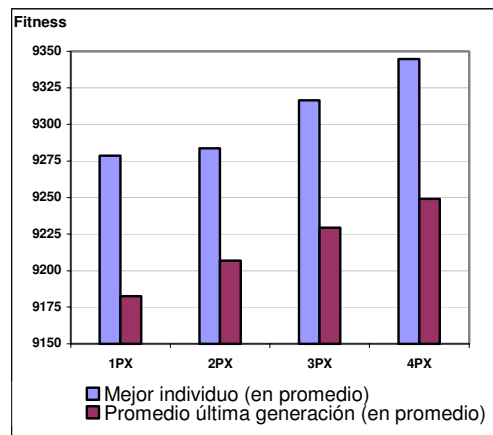


Figura 7: Resultados de operadores de n puntos, grafo 50-15.

Para el grafo de 50 nodos y 15 terminales no se percibe una relación directa entre la disrupción y la calidad de resultados al utilizar operadores de cruzamiento uniforme. Los resultados obtenidos para las diferentes parametrizaciones son muy similares, ubicándose sus diferencias por debajo de la desviación estándar de las ejecuciones realizadas, por lo que no pueden considerarse significativas.

Los operadores de  $n$  puntos presentan el comportamiento detectado para los restantes grafos, mejorando la calidad de resultados al considerar operadores más disruptivos.

Para este grafo, no se verifica una superioridad en la calidad de los resultados de la familia de uniformes parametrizados sobre la de  $n$  puntos.

En lo referente a tiempos de ejecución, no se advierten grandes diferencias entre las variantes de cada familia para los grafos estudiados. Se observa que los operadores uniformes en general detienen su evolución en un número de generaciones cercano al 3/4 del tope establecido, luego del cual no se logra mejorar los resultados obtenidos. Los operadores de  $n$  puntos evolucionan mejorando los resultados hasta un número mayor de generaciones, salvo para el grafo de 50 nodos y 15 terminales, donde exhiben un comportamiento similar a los uniformes.

## **6. Conclusiones y trabajo futuro**

### **6.1 - Conclusiones**

Los resultados obtenidos no pueden considerarse concluyentes debido a que en varios casos no presentan diferencias significativas en la calidad de resultados para los grafos considerados. De todos modos, es posible indicar que para la resolución del problema GSP con la codificación binaria utilizada, en general las estrategias más disruptivas obtienen los mejores resultados. Además, la familia de cruzamiento uniforme parametrizado obtiene resultados de mejor calidad que los obtenidos por la familia de cruzamiento de  $n$  puntos cuando se utiliza el criterio de parada de esfuerzo prefijado en 2000 generaciones. Dentro de los cruzamientos uniformes parametrizados, en general los mejores resultados se obtienen al trabajar con un parámetro de intercambio alto, entre 0.4 y 0.5. Para los cruzamientos de  $n$  puntos los mejores resultados se obtuvieron al utilizar la mayor cantidad de puntos de corte.

En lo referente a tiempos de ejecución, no se detectaron diferencias significativas que permitan concluir que el uso de una familia o de un operador en particular involucre un mayor tiempo de uso de recursos computacionales. La evolución de los operadores uniformes se detiene antes que los operadores de  $n$  puntos, por lo cual podrían existir diferencias en cuanto a los tiempos de ejecución, si se utilizara un criterio de parada basado en la detección de convergencia.

### **6.2 - Investigación actual y futura**

Las líneas de investigación actual y futura se concentran en completar el estudio sobre el efecto de los operadores de cruzamiento sobre el AG aplicado al GSP, incluyendo los operadores de cruzamiento adaptativos no analizados en este trabajo. Estos operadores tienen como idea básica aprender características del problema para mejorar la calidad de los resultados y la propia técnica del cruzamiento. Este camino se presenta como prometedor para mejorar la calidad de los resultados obtenidos.

Complementariamente, debe estudiarse la escalabilidad de los resultados obtenidos para las instancias particulares analizadas en este trabajo. Al considerar grafos representativos de redes de comunicaciones de mayor tamaño que los casos estudiados, será posible determinar la validez de las conclusiones obtenidas en este trabajo.

El estudio de los operadores de cruzamiento se realizó sobre modelos secuenciales de AG. Teniendo en cuenta el elevado costo computacional de los algoritmos, la extensión del análisis para modelos distribuidos constituye una interesante línea de trabajo. Debido a que el modelo de evolución de los AG distribuidos es diferente al de los secuenciales, verificar la extensión de los resultados obtenidos en este trabajo constituirá en sí mismo un resultado importante.

## 7. Agradecimientos

Este trabajo fue desarrollado en el marco del proyecto "Algoritmos Genéticos Paralelos y su Aplicación al Diseño de Redes de Comunicaciones Confiables", financiado por la Comisión Sectorial de Investigación Científica de la Universidad de la República. Agradecemos el apoyo del Dr. Héctor Cancela del Departamento de Investigación Operativa, Instituto de Computación de la Facultad de Ingeniería, Universidad de la República, Uruguay y del Dr. Enrique Alba, del Departamento de Lenguajes y Ciencias de la Computación de la Universidad de Málaga, España.

## Referencias bibliográficas

- [1] E. Alba, F. Almeida, M. Blesa, C. Cotta, M. Díaz, I. Dorta, J. Gabarró, J. González, C. León, L. Moreno, J. Petit, J. Roda, A. Rojas, F. Xhafa, *MALLBA: A Library of Skeletons for Combinatorial Optimisation*, In Proceedings of the EuroPar, pp. 927-932, 2002.
- [2] M. Aroztegui, S. Árraga, S. Nesmachnow, *Resolución del Problema de Steiner Generalizado utilizando un Algoritmo Genético Paralelo*, II Congreso Español sobre Metaheurísticas, Algoritmos Evolutivos y Bioinspirados, pp. 387-394, 2003.
- [3] K. De Jong, *An Analysis of the Behavior of a Class of Genetic Adaptive Systems*, PhD Thesis, Dept. Computer & Communication Sciences, Univ. of Michigan, 1975.
- [4] H. Esbensen, *Computing Near-Optimal Solutions to the Steiner Problem in a Graph Using a Genetic Algorithm*, Networks 26, pp. 173-185, 1995.
- [5] L. Ford, D. Fulkerson, *Flows in Networks*, Princeton University Press, 1962.
- [6] J. Holland, *Adaptation in Natural and Artificial Systems*, MIT Press, 1975.
- [7] V. Kahn, P. Crescenzi, *A Compendium of Optimization Problems*, Disponible online: <http://www.nada.kth.se/theory/problemlist.html>. Consultado julio 2003.
- [8] R. Karp, *Reducibility among combinatorial problems*, Complexity of Computer Communications, Plenum Press, pp. 85-103, 1972.
- [9] F. Robledo, *Diseño Topológico de Redes, Casos de Estudio: The Generalized Steiner Problem y The Steiner 2-Edge-Connected Subgraph Problem*, Master Tesis, PEDECIBA Informática, Universidad de la República, Montevideo, Uruguay, 2001.
- [10] W. Spears, *Recombination Parameters*, en The Handbook of Evolutionary Computation, Oxford University Press, 1997.
- [11] W. Spears, K. De Jong, *An Analysis of Multi-Point Crossover*, Proceedings of Foundations of Genetic Algorithms Workshop, pp. 301-315, 1991.
- [12] W. Spears, K. De Jong, *On the Virtues of Parameterized Uniform Crossover*, Proceedings of the 4th Int. Conference on Genetic Algorithms, pp. 230-236, 1991.
- [13] G. Syswerda, *Uniform Crossover in Genetic Algorithms*, Proceedings of the 3rd Int. Conference on Genetic Algorithms, pp. 2-9, 1989.